Technical Documentation API

Money Transfer Hub provides possibility to process Person-2-Person and Person-2-Merchant transactions in various forms. Please check details in the below documentation.

> This documentation contains the methods for **mobile-server** integration. The methods included in the documentation are intended for Customers creating their own mobile SDK.

> The Customer creating the SDK must also remember about the integration with the MobileDC component.

Receiver types which can be used to set Receiver.Type

Based on ReceiverType user can fill different field in Receiver object in requests.

ReceiverType	Description
BARE_CARD_NUMBER	Bare card number in **Receiver.card** field
FRIEND_ID	Should pass FriendId in **Receiver.Card** field
WALLET_CARD_ID	Should pass DataCoreCardId to **Receiver.Card** field and UserDataCoreCardId to **Receiver.userId** field
EMPTY	Means that the receiver have the same card data like sender. This type may be useful on Determine Currency

JWE

Peer To Peer Transaction Service supports encryption of requests and responses as standard JSON Web Encryption (JWE) per RFC 7516.

Recommended to read the JWE standard: RFC 7516.

Methods that support request encryption in the JWE standard are tagged in the documentation with the header: *Content-Type:application/x-jwe-encryption-body+json*. If the response is to be encrypted with the JWE standard then the header must be added: *X-Encryption-Public-Key* with the

public key.

Processing requests and responses can be divided into 4 options listed below:

1. Base request → Base response - the following headers should be provided to pass this case:
   - *Content-Type: application/json*
2. Base request → Encrypted response - the following headers should be provided to pass this case:
   - *Content-Type: application/json*
3. Encrypted request → Base response - the following headers should be provided to pass this case:
   - *Content-Type: application/x-jwe-encryption-body+json*
4. Encrypted request → Encrypted response - the following headers should be provided to pass this case:
   - *Content-Type: application/x-jwe-encryption-body+json*

Overview

JWE represents encrypted content using JSON data structures and Base64 encoding. The representation consists of three parts: a JWE Header, a encrypted payload, and a signature. The three parts are serialized to UTF-8 bytes, then encoded using base64url encoding. The JWE's header, payload, and signature are concatenated with periods (.).

JWE typically takes the following form:

```
{Base64 encoded header}.{Base64 encoded payload}.{Base64 encoded signature}
```

JWE header contains:

Type	Value	Constraints	Description
alg	RSA-OAEP-256	Required	Identifies the cryptographic algorithm used to secure the JWE Encrypted Key. Supported algorithms: **RSA-OAEP-256**, **RSA-OAEP-384**, **RSA-OAEP-512**. Recommend value: **RSA-OAEP-256**.

enc	A256GCM	Required	Identifies the cryptographic algorithm used to secure the payload. Supported algorithms: **A128GCM**, **A192GCM**, **A256GCM**, **A128CBC-HS256**, **A192CBC-HS384**, **A256CBC-HS512**. Recommend value: **A256GCM**.
typ	JOSE	Optional	Identifies the type of encrypted payload. Recommend value: **JOSE**.
iat	1637929226	Optional	Identifies the time of generation of the JWT token. Supported date format: unix time in UTC. In the case of *iat* send, the validity of JWE is validated. Recommend send the header due to the increase in the security level.
kid	5638742a5094327fcd7a5945d06a45a9d83e9006	Optional	Identifies the public key of use to encrypt payload. Supported format: SHA-1 value of the public key. In the case of *kid* send, the validity of public key is validated, so we can inform the client that the public key has changed.

Payload Encryption

Every encrypted request should include JWE token. The jwe token should be passed in the field: *value*.

In case of problems with the implementation of JWE, please contact the administrator.

To prepare the encrypted payload:

The steps may differ depending on the libraries used.

1. Get the public key using the method: [???](#Get publicKey). The public key is encoded with Base64.
2. Decode the public key.
3. Then create a correct object to be encrypted.
4. Encrypt the created object with the public key.
5. Create JWE header compatible with: JWE Header

6. Make a request on the method that supports JWE. Set the JWE token in the field: *value*. Methods supporting JWE use the following header: *Content-Type:application/x-jwe-encryption-body+json*.

Payload Decryption

To prepare the decrypted payload:

The steps may differ depending on the libraries used.

The cryptographic algorithm used to secure the payload is: **A256GCM**, while to secure the encrypted JWE key: **RSA-OAEP-256**.

1. For the response to be encrypted you need to send *public key* in the header: *X-Encryption-Public-Key*. The header value must be encoded *Base64*.
2. After receiving the response, you should get the JWE token from the field: *value*.
3. Decrypt the JWE token from the field: *value* with the private key.

Public key format to be encoded in Base64.

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0IDAQAB...
-----END PUBLIC KEY-----
```

P2P open API

@swagger="https://p2ptransactions.upaidtest.pl/docs/index.yaml"

Revision #39
Created 15 March 2022 13:27:30
Updated 11 March 2025 10:28:35